

# Estructuras de Datos 1 - ST0245

## Examen Parcial 2

Nombre:.....  
Departamento de Informática y Sistemas  
Universidad EAFIT

Mayo 9, 2017

### Criterios de calificación

- Selección múltiple con única respuesta
  - Respuesta correcta: 100 %
  - Respuesta incorrecta: 0 %
- Completar código
  - Respuesta correcta 100 %
  - Respuesta incorrecta o vacía 0 %

#### NOTAS IMPORTANTES:

- Responda en la hoja de PREGUNTAS
- Marque la hoja de PREGUNTAS

### 1. Árboles binarios 20 %

En este parcial, el nodo de un árbol binario se define así:

```
class Nodo {  
    Nodo izq;  
    Nodo der;  
    int dato;  
}
```

Kefo ha dañado a Dayla su código para determinar **cuál es la altura máxima de un árbol binario**. Dayla no recuerda como lo había hecho y te pide ayuda para que lo completes.

```
01 int altura(Nodo raiz){  
02     if(raiz == null)  
03         return 0;  
04     int izq =_____  
05     int der =_____;
```

```
06     return Math.max(izq, der);};
```

A (10 %) Complete el espacio en línea 04

-----  
B (10 %) Complete el espacio en línea 05

### 2. Árbol bin. de búsqueda 10 %

A un árbol binario de búsqueda se le ingresan 9 elementos en este orden: 12, 3, 5, 8, 2, 1, 9, 17. **¿Cuántos nodos hay que recorrer antes de encontrar el número 8?** Se cuenta la raíz, pero no se cuenta el nodo donde está el 8.

- A 2
- B 1
- C 3
- D 5

### 3. Recursión 30 %

Dayla y Kefo están aquí de nuevo. En esta vez han traído un juego muy interesante, en el cual Kefo, en primer lugar, escoge un número  $n$  ( $1 \leq n \leq 20$ ) y, en segundo lugar, escoge tres números  $a, b$  y  $c$  ( $1 \leq a \leq 9, 1 \leq b \leq 9, 1 \leq c \leq 9$ ). Después, Kefo le entrega estos números a Dayla y Dayla le tiene que **decir a Kefo la cantidad máxima de números tomados de  $a, b$  y  $c$  (se puede tomar un número más de una vez) que al sumarlos dan el valor  $n$ .**

Como un ejemplo, si Kefo escoge  $n = 14$  y  $a = 3, b = 2, c = 7$ . ¿Qué posibilidades hay de obtener 14 con  $a, b$  y  $c$ ?

$7 + 7 = 14$	cantidad es 2
$7 + 3 + 2 + 2 = 14$	cantidad es 4
$3 + 3 + 3 + 3 + 2 = 14$	cantidad es 5
...	
$2 + 2 + 2 + 2 + 2 + 2 + 2 = 14$	cantidad es 7

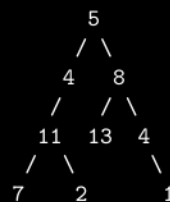
La cantidad máxima de números es 7. Esta sería la respuesta que da Dayla a Kefo. Como Dayla es muy astuto, ha diseñado un algoritmo para determinar la cantidad máxima de números y quiere que le ayudes a terminar su código. Asuma que hay al menos una forma de sumar  $n$  usando los números  $a, b$  y  $c$  en diferentes cantidades, incluso si algunos de los números se suman 0 veces como sucede en el ejemplo anterior.

```
1 int solucionar (int n, int a, int b, int c) {
2   if (n == 0 || (n < a && n < b && n < c))
3     return 0;
4   int res = solucionar(_____) + 1;
5   res = Math.max(_____, _____);
6   res = Math.max(_____, _____);
7   return res; }
```

- A (10%) Complete el espacio de la línea 04  
 -----  
 B (10%) Complete los espacios de la línea 05  
 -----, -----  
 C (10%) Complete los espacios de la línea 06  
 -----, -----

## 4. Árboles binarios 40%

Definimos el camino desde la raíz hasta una hoja en un árbol binario como una secuencia de nodos empezando en el nodo raíz y bajando hasta una hoja. Una hoja es un nodo que no tiene hijos. Decimos que un árbol vacío no contiene caminos desde la raíz hasta una hoja. Como un ejemplo, el siguiente árbol tiene 4 caminos desde la raíz hasta una hoja:



Caminos desde la raíz hasta una hoja:

- camino 1: 5 4 11 7
- camino 2: 5 4 11 2
- camino 3: 5 8 13
- camino 4: 5 8 4 1

Para este problema nos interesan las sumas de los elementos en los nodos de esos caminos, por ejemplo, la suma del camino [5, 4, 11, 7] es 27. Dada la raíz de un árbol binario `Nodo a` y un entero `int suma`, decir si existe un camino desde la raíz hasta una hoja tal que al sumar los valores de los nodos de ese camino la suma sea igual al parámetro `suma`. Retorne falso si no se puede encontrar un camino con esa condición. Utilice la definición de `Nodo` del punto 1.

Desafortunadamente, al código que hizo Dayla le faltan unas líneas y Kefo está de vacaciones.

```
1 boolean sumaElCamino(Nodo a, int suma) {
2   if (a == null)
3     return _____;
4   if (a.izq == null && a.der == null)
5     return suma == _____;
6   else
7     return sumaElCamino(_____, _____)
8       || sumaElCamino(_____, _____); }
```

- A (10%) Complete el espacio de la línea 03  
 -----  
 B (10%) Complete el espacio de la línea 05  
 -----  
 C (10%) Complete los espacios de la línea 07  
 -----, -----  
 D (10%) Complete los espacios de la línea 08  
 -----, -----